# Introduction au calcul scientifique avec Matlab

## TP2

## 1. Introduction

Pour effectuer ce TP, si vous n'avez pas accès à une licence Matlab sur votre machine, vous pouvez utiliser le logiciel GNU Octave qui est un clone gratuit et libre de Matlab. En règle générale les codes se portent facilement de Matlab vers Octave et vice-versa.

L'évaluation du TP tiendra compte de la rédaction du rapport, qualité des solutions mathématiques, pertinence des commandes utilisées (pensez chemin le plus court pour arriver aux résultats).

Le journal de bord de votre session doit être joint au rapport de la séance.
Tous les scripts et fichiers de fonctions doivent être rangés dans un seul document word ou txt.

## 2. Les fonctions

### 2.1 Les fonctions élémentaires

Les fonctions élémentaires de Matlab s'appliquent à des scalaires. Lorsqu'on les applique à des vecteurs ou matrices, elles opèrent terme par terme.
Quelques fonctions élémentaires :

| | | |
|---|---|---|
| `sqrt` | `exp` | `log` |
| `sin` | `cos` | `tan` |
| `asin` | `acos` | `atan` |
| `round` | `floor` | `ceil` |
| `abs` | `angle` | `conj` |

### 2.2 Les fonctions vectorielles

Les fonctions vectorielles de Matlab s'appliquent à l'ensemble d'un vecteur. Lorsqu'on les applique à des matrices, elles opèrent colonne par colonne.
Quelques fonctions vectorielles :

| | |
|---|---|
| `max(x)` | maximum |
| `min(x)` | minimum |
| `sort(x)` | tri par ordre croissant |
| `[y, I] = sort(x)` | retourne en plus les indices des élements de `x` |
| `find(x)` | retourne les indices non nuls de `x` |
| `[y, I] = find(x)` | retourne des lignes (dans le vecteur `I`) et des colonnes (dans le vecteur `J`) des élements non nuls du `x` |
| `sum(x)` | somme des éléments de `x` |
| `cumsum(x)` | vecteur contenant la somme cumulée des éléments de `x` |
| `prod(x)` | produit des éléments de `x` |
| `cumprod(x)` | vecteur contenant le produit cumulé des éléments de `x` |
| `diff(x)` | vecteur des différences entre deux éléments consécutifs de `x` |
| `mean(x)` | moyenne des éléments de `x` |
| `median(x)` | médiane |
| `std(x)` | écart type |

## 3. Les opérateurs relationnels et logiques

| | |
|---|---|
| `Opérateurs relationnels` | `<`, `<=`, `>=`, `==` (égalité), $\sim$`=` (différent) |
| `Opérateurs logiques` | & (et), \| (ou), $\sim$ ou `not` (non) |

Les opérateurs relationnels peuvent être utilisés avec des scalaires ou des matrices. Le résultat d'évaluation d'une expression relationnelle est 1 (vrai) ou 0 (faux). Appliqués à une matrice, ils rendent une matrice de même dimension, formée de 1 et de 0.

Remarque : Attention de ne pas confondre = qui sert à affecter une valeur à une variable et == qui sert à tester l'égalité.

## 4. Exercices

Exercice 1.

Observer l'effet des instructions suivantes.

```
>> x=rand(1,5)
>> mean(x)
>> std(x)
>> median(x)
>> sort(x)
>> A=rand(3)
>> sort(A)
>> [B, I]=sort(A)
>> sort(A')
>> max(A)
>> max(A')
>> max(max(A))
>> sum(A)
>> cumsum(A)
>> prod(A)
>> diff(A)
>> D=A([1,2],1:3)
>> sum(D,1)
>> sum(D,2)
```

Exercice 2.

Soit X une matrice 2xn contenant les coordonnées de n points du plan. Comment faire pour obtenir une matrice où les points sont ordonnés par ordre croissant des abscisses ? (sans utiliser les boucles)

Exercice 3.

1. Soit le vecteur de dimension 8 de composantes : 3.2; 4.8; 3.3; 3.2; 3.1; 4.2; 3.2; 3.3 Construisez le vecteur y = $(y_i)_{i=1;...;8}$ correspondant.
2. Construisez, à l'aide des fonctions vectorielles (sans utiliser les boucles), une fonction *moyenne.m* qui a comme argument d'entrée n et qui permet de calculer la suite des moyennes mobiles:

$$\bar{y}_n = \frac{1}{n}\sum_{i=1}^{n} y_i \ .$$

3. Donner la valeur de $\bar{y}_8$.

Exercice 4.

Soit la matrice A :

```
A=[1  2  3;  2  3  1;  3  1  2]
```

Construisez, à l'aide des opérateurs logiques, la fonction *find.m* qui a pour argument d'entrée un scalaire k et qui permet de retourner les termes égaux à k ainsi que leur position.

Exercice 5.

Reprenez le vecteur y de l'exercice 3.

1. Ecrire une fonction *existe.m* qui a comme argument un vecteur quelconque z et un scalaire k et qui permet de retourner la réponse à la question suivante : existe-t-il une coordonnée du vecteur z inférieur à k.
2. Utiliser la fonction existe.m pour répondre à la question suivante : existe-t-il une coordonnée du vecteur y inférieure à 3.3

Ecrire un script qui permet de construire un vecteur logique z tel que la ième coordonnée de ce vecteur sera 1 si la ième coordonnée du vecteur y est à l'extérieur de l'intervalle :

$$[\bar{y}_8 - \sigma_8, \bar{y}_8 + \sigma_8]$$

Où $\sigma_8$ est la racine carrée de la variance d'échantillonnage.